UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/543,930	04/06/2000	John David Colleran	202732	8589

7590 05/21/2004

Leydig Voit & Mayer LTD
Two Prudential Plaza
Suite 4900
180 North Stetson
Chicago, IL 60601-6780

EXAMINER	
LEWIS, ADAM M	
ART UNIT	PAPER NUMBER
2174	6

DATE MAILED: 05/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/543,930	COLLERAN ET AL.
	Examiner	Art Unit
	Adam M. Lewis	2174

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no avent, howaver, may a reply be timely filed after SIX (6) MONTHS from tha mailing date of this communication.
- If tha period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within tha set or extended period for reply will, by statute, causa tha application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after tha mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 March 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *16-27* is/are allowed.

6)☒ Claim(s) *1-15,28-30 and 32-35* is/are rejected.

7)☒ Claim(s) *31* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This communication is responsive to Amendment A, filed 4 March, 2004.

2. Claims 1-35 are pending in this application. Claims 1, 16, 28, 30, and 35 are independent claims. In Amendment A, claims 1-4, 6-16, and 18-35 were amended. This action is made Final.

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-15 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: placing a high priority special event in an event queue and detecting a wakeup message and responsively replacing the ghost user interface with the original user interface.

Based on the specification it is apparent that the omitted elements mentioned above are essential to the functionality of the invention, however the amendment has cancelled those limitations, leaving the claim indefinite and contrary to the specified invention.

Claim Rejections - 35 USC § 102

6. Claims 1, 3, and 35 are rejected under 35 U.S.C. 102(b) as being anticipated by Boston ("Boston," US# 5,515,493).

As per independent claim 1, Boston teaches a method for managing an original user interface for an application, the method comprising:

signaling a hung state for the application (Boston, col. 3,lines 64-66);

creating a ghost user interface in response to the hung state that appears in place of the original user interface (Boston, col. 4, lines 13-28);

detecting activity in an event queue of the application (Boston, col. 4, lines 59-61); and

replacing the ghost user interface with the original user interface in response to the detection of activity (Boston, col. 4, lines 61-65).

As per claim 3, which is dependent on claim 1, Boston teaches the method of claim 1 further comprising releasing resources used by the ghost user interface in response to detecting activity in the event queue (inherent in Boston, col. 4, lines 61-65). When deleting windows in a GUI the resources used by those windows are automatically released back to the operating system.

As per independent claim 35, Boston teaches a computer-readable media having computer executable instructions for carrying out the steps of a method for managing an original user interface for an application, the method comprising:

signaling a hung state for the application (Boston, col. 3, lines 64-66);

creating a ghost user interface in response to the hung state that appears in place of the original user interface (Boston, col. 4, lines 13-28);

detecting activity in an event queue of the application (Boston, col. 4, lines 59-60); and

replacing the ghost user interface with the user interface in response to the detection of activity (Boston, col. 4, lines 59-65).

Claim Rejections - 35 USC § 103

7. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Boston.

As per claim 2, which is dependent on claim 1, Boston fails to teach the limitation of signaling a hung state occurs if the application does not handle an event in the event queue for at least a predetermined duration. However, OFFICIAL NOTICE is given that using a timeout to determine lack of activity is well known in the art. It would have been obvious to one skilled in the art at the time of invention to utilize a timeout to determine non-responsiveness in the invention of Boston because it would increase efficiency and have very little programming overhead.

8. Claims 4-6, 28-30, and 32-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boston in view of Silberschatz ("Silberschatz," *Operating System Concepts*, ISBN 0-201-50480-4, 1994).

As per claim 4, which is dependent on claim 1, Boston fails to teach the method of claim 1 wherein creating the ghost user interface includes creating a ghost thread for the ghost user interface. However, Silberschatz teaches that threads provide a mechanism that allows sequential processes to make blocking system calls while also achieving parallelism (Silberschatz, Page 112, ¶ 4). This makes the process more

efficient as well as allowing the ghost thread to access the GUI thread without the overhead of programming around the protections included in the GUI thread (Silberschatz, Page 112, ¶ 2). It would have been obvious to one skilled in the art at the time of invention to include a thread specifically for the ghost interface in the ghost interface system of Boston because it would increase the efficiency of the process as well as reduce the amount of programming required.

As per claim 5, which is dependent on claim 4, Boston fails to teach the method of claim 4 wherein creating the ghost thread includes determining if a ghost thread exists and creating a ghost thread if the ghost thread does not exist. However, OFFICIAL NOTICE is given that checking for the existence of an executing code segment is well known in the art. It would have been obvious to one skilled in the art at the time of invention to include ensuring a ghost thread did not exist before creating one in the invention of Boston and Silberschatz because it would increase the overall efficiency of the GUI.

As per claim 6, which is dependent on claim 4, Boston further teaches the method of claim 4 wherein creating the ghost user interface includes creating the ghost user interface in the area occupied by the original user interface (Boston, col. 4, lines 1-6; col. 4, lines 13-29).

As per independent claim 30, Boston teaches a multithreaded computing system for executing an application having a user interface, wherein if the application is non-responsive to user input, the user interface is replaced by a ghost user interface, the system comprising:

a non-responsive application detecting code for detecting a non-responsive application (Boston, col. 1, lines 64-67);

creating the ghost user interface to replace the user interface in response to detection of a non-responsive application (Boston, col. 4, lines 13-29);

a high priority special entry in a queue for the application for detecting if the application is responsive (Boston, col. 4, lines 59-64); and

a plurality of computer executable instructions for destroying the ghost user interface in response to handling of the high priority special entry by the application (Boston, col. 4, lines 59-64). Boston fails to teach the limitation of a specific ghost thread.

However, Silberschatz teaches that threads provide a mechanism that allows sequential processes to make blocking system calls while also achieving parallelism (Silberschatz, Page 112, ¶ 4). This makes the process more efficient as well as allowing the ghost thread to access the GUI thread without the overhead of programming around the protections included in the GUI thread (Silberschatz, Page 112, ¶ 2). It would have been obvious to one skilled in the art at the time of invention to include a thread specifically for the ghost interface in the ghost interface system of Boston because it would increase the efficiency of the process as well as reduce the amount of programming required.

Independent claim 35 is similar in scope to claim 30, and is therefore rejected under similar rationale.

As per claim 32, which is dependent on claim 30, the invention of Boston and Silberschatz fails to teach the system of claim 30 further comprising a second non-responsive application with a second user interface and a second ghost user interface created by the ghost thread. However OFFICIAL NOTICE is given that multiple applications running and/or hanging within a GUI is well known in the art. It would have been obvious to one skilled in the art at the time of invention to include functionality to repeat the ghost user interface process for a second window into the invention of Boston and Silberschatz because it would allow the user to continue using the system regardless of how many applications became non-responsive.

Independent claim 28 is similar in scope to claim 32, and is therefore rejected under similar rationale. The flip- and flop- window signals correspond to detecting a hung application and detecting the hung application becoming responsive again, as noted in applicants specification (Page 25, lines 14-17).

As per claim 33, which is dependent on claim 30, Boston further teaches the system of claim 30 further comprising a responsive-application detecting code for detecting when a non-responsive application becomes responsive (Boston, col. 4, lines 59-64).

As per claim 34, which is dependent on claim 33, Boston further teaches the system of claim 33 further comprising a responsive-application user interface restoring code for replacing the first ghost user interface with the first user interface in response to detecting that the first application has become responsive (Boston, col. 4, lines 59-64).

Dependent claim 29 is similar in scope to claim 34, and is therefore rejected under similar rationale.

Allowable Subject Matter

9. Claims 16-27 are allowed.

10. Claims 7-15 and 31 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

11. The following is a statement of reasons for the indication of allowable subject matter: The patentable distinction of claims 7 and 16 over the prior art is the limitation of directing and caching input, respectively, corresponding to the substitute user interface element wherein the directed or cached input is subsequently handled by the application. While the prior art teaches the substitute user interfaces capability to intercept and act upon operating system level instructions such as repaints, the prior art fails to disclose handling and/or storing input for further use.

Response to Arguments

12. Applicant's argued the following:

a. Applicant asserts a distinction between a "halted" application vs. a "hung" application. Boston states in part, "...the debugger is responsive to the debuggee application (DA) being halted (Block A), at the end of a step or run command..." (col. 3, lines. 64-67). This statement implies that the DA was in a halted state prior to the

intervention of the debugger, contrary to Applicant's assertion. Furthermore, it is well known in the debugger art that a debugger may run in two modes, step or run (Boston '493: col. 3, line. 67), in which the latter mode waits for an interrupt or an exception as signaled by the operating system, in order to determine if a debugging operation should commence. In other words, contrary to Applicant's assertion, the notification of the DA's hung state is arriving from the operating system, rather than the debugger itself.

b. Applicant asserts that the Boston does not "detect activity in an event queue of the application" (Amendment A: p. 13, lns. 5-6). Examiner disagrees as follows. It is well known in the art, that GUI applications, run "message pumps" i.e. infinite loops to process events as dispatched by the operating system. Specifically, the operating system receives an event, such as a mouse click, identifies the relevant or active application, queues the event, and dispatches the event to the relevant or active application's message pump. The application message pump, subsequently "handles" said event. If the event handler throws an exception or causes a fault (e.g. access violation, division by zero, or any other well known error conditions), it will notify the operating system, and will provide the debugger with a STACK trace. This stack trace includes a pointer that indicates the offending event handler. Therefore, contrary to the applicant's assertion, by virtue of receiving a pointer to the event handler, the debugger of Boston is in fact "detecting activity in the event queue, of the application."

c. Applicant asserts that Boston cannot disclose the step of "replacing the ghost interface with the original user interface in response to the detection of activity" by virtue of the fact that Boston does not detect activity in an event queue of the application.

However, since it has been shown above that Boston can, and does, detect activity in an event queue of the application, Boston can, as cited in the rejection, replace the ghost interface with the original user interface.

13. Applicant's arguments with respect to claims 28 and 29 have been considered but are moot in view of the new ground(s) of rejection.

14. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Adam M. Lewis whose telephone number is 703-305-0720. The examiner can normally be reached on M-Th 7:00-4:30, Alternate Fridays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine L. Kincaid can be reached on 703-308-0640. The fax phone

number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kristine Kincaid

KRISTINE KINCAID
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

al